https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

Machine learning based radiation emulation: development and performance evaluation in operational reforecast experiments

Hao Jing[1,2], Sa Xiao[1,2], Haoyu Li[1,2], Huadong Xiao[1,2], Wei Xue[2,3]

[1]State Key Laboratory of Severe Weather Meteorological Science and Technology (LaSW), Beijing, 100081, China
[2]CMA Earth System Modeling and Prediction Centre (CEMC), Beijing, 100081, China
[3]Tsinghua University, Beijing, 100081, China

Correspondence to: Sa Xiao (xiaos@cma.gov.cn)

Abstract. Radiation is typically the most time-consuming physical process in numerical models. One solution is to use machine learning methods to simulate the radiation process to improve computational efficiency. From a practical application standpoint, this study investigates the key issues for hybrid modelling: coupling compatibility and long-term integration stability. A residual convolutional neural network is employed to approximate the Rapid Radiative Transfer Model for General Circulation Models (RRTMG) within the global operational system of China Meteorological Administration. We adopted an offline training and online coupling approach. First, a comprehensive dataset is generated through model simulations, encompassing all atmospheric columns. To ensure the stability of the hybrid model, the dataset is enhanced via experience replay, and additional output constraints based on physical significance are imposed. Meanwhile, a LibTorch-based coupling method is utilized, which is more suitable for real-time operational computations. The hybrid model is capable of performing ten-day integrated forecasts as required. A two-month operational reforecast experiment demonstrates that the machine learning emulator achieves accuracy comparable to that of the traditional physical scheme, while accelerating the computation speed by approximately eightfold.

1 Introduction

In recent years, machine learning methods have exhibited exceptional performance in addressing problems characterized by high dimensionality and large datasets, while also bringing new vitality and challenges to the field of numerical weather prediction. Currently, meteorological operational and research institutions worldwide have initiated relevant research programs to explore the application of artificial intelligence (AI) technology in weather and climate prediction (ECMWF, 2025). Existing studies have integrated AI methods across the entire workflow, including observation operators (Liang et al., 2023), satellite and radar data processing (Ukamaka et al., 2024), physical parameterization (Chantry et al., 2021; Gao et al., 2024), data assimilation (Arcucci et al., 2021), ensemble forecasting (Zhong et al., 2025), and post-processing (McGovern et al., 2017). Furthermore, a growing body of research has applied AI technology to improve weather forecasts in fields like precipitation (Chen et al., 2023), tropical cyclones (Wu et al., 2022), and typhoons (Niu et al., 2024). These methods have demonstrated significant potential to enhance prediction accuracy and boost computational efficiency.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

Machine learning technology performs well in identifying and learning complex nonlinear relationships within datasets (Schmidhuber, 2015). Theoretically, deep neural networks possess the capability to approximate any function (Cybenko, 1989). It implies that they can also model the equations or functions describing atmospheric motion. With the increasing adoption of machine learning in atmospheric science, researchers have explored the development of new parameterization schemes. These schemes leverage machine learning to balance computational efficiency and accuracy, enabling them to not only accurately identify and extract complex nonlinear relationships inherent in physical processes but also address the bottleneck of excessively high computational costs faced by numerical model development. The use of neural network models to replace or optimize physical process parameterization schemes falls under the category of general form regression tasks. Specifically, neural networks are trained to learn the relationships between sets of input and output vector pairs. Once trained, they are coupled with numerical models to achieve model replacement or optimization. Such studies can be divided into two categories. The first aims to simulate existing physical parameterization schemes, leveraging the high-efficiency fitting capability of neural networks to reduce computational costs. The second improves physical parameterization schemes by incorporating additional information. For instance, using observation data from satellites, radars and other platforms, or training on higher-resolution models to develop new models to enhance computational accuracy. This hybrid modeling approach (Reichstein et al., 2019), which integrating numerical methods with data-driven techniques, exemplifies the trend and impact of interdisciplinary collaboration across artificial intelligence, high-performance computing, and numerical weather prediction.

Over two decades ago, studies begun to use trained neural networks for simulating radiation parameterization schemes. For example, Chevallier et al. (2000) employed a multi-layer perceptron to replicate the radiation scheme of the ECMWF Integrated Forecast System (IFS) model, while Krasnopolsky et al. (2005) developed a neural network model for application in the National Center for Atmospheric Research (NCAR) CAM2 model. These efforts yielded results comparable to the original radiative transfer processes and increased computational speed by an order of magnitude. However, the neural network structures used at that time were simplistic, preventing full simulation of the entire radiation process. In recent years, advances in deep learning technology have enabled the successful development of deep learning based parameterization schemes for radiation using state-of-the-art techniques. Song et al. (2021) developed two compound radiative parameterizations to reduce uncertainty arising from the extrapolation of extreme data beyond training sets. The first leverages an additional neural network to predict heating rate errors for all input variables, while the second uses cloud fraction to estimate the uncertainty of the neural network emulation. Both approaches significantly reduced the forecast error of the hybrid model. Ukkonen (2022) demonstrated that bidirectional recurrent neural networks can natively incorporate the vertical dependencies of radiation computations. By mimicking the structural flow of physical radiative transfer solvers, the recurrent neural network achieves substantially higher accuracy than standard feed-forward neural network while utilizing an order of magnitude fewer parameters. Alves et al. (2023) incorporated 3D cloud radiative effects into ecRad (Hogan and Bozzo, 2016, a library used by ECMWF to compute vertical profiles of solar and near-infrared fluxes and heating rates) by testing various model architectures and different coupling methods. Hafner et al. (2025) developed a Bidirectional Long

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

Short-Term Memory (BiLSTM) emulator for the ICON model and utilized Shapley Additive exPlanations to interpret the model's decision-making process. Their analysis successfully demonstrated that physics-inspired architectures like BiLSTMs inherently capture causal physical mechanisms – such as the non-local cooling effect of cloud reflection in shortwave radiation and local temperature dependencies in longwave emission.

On the other hand, deep learning based physical parameterization schemes still face critical issues related to stability and physical interpretability. Coupled operation denotes a framework in which the deep learning parameterization module receives inputs from the numerical model, generates predictions, and feeds outputs back to other model components. This procedure proceeds iteratively to support time-integrated simulations. Stability during coupled operation is not only a prerequisite for long-term integration but also a key factor in facilitating the practical deployment of hybrid models in operational numerical prediction systems. Numerical instability in this process may stem from non-physical correlations (Brenowitz and Bretherton, 2019), inadequate enforcement of physical constraints (Yuval et al., 2021), and suboptimal selection of model network structures, hyperparameters, and optimizers. Chen et al. (2025) developed a machine learning based physical parameterization scheme trained with experience replay, which enabled stable long-term simulations in the Community Atmosphere Model. The key mechanism lies in dynamically filtering training samples to mitigate error accumulation, offering a new training paradigm for enhancing the stability of hybrid models. As network depth increases, the complexity and abstraction of deep learning models grow rapidly. At present, no comprehensive theory exists to systematically interpret their internal decision-making logic (Hernandez-Orallo, 2021), resulting in limited transparency and interpretability. Such insufficient interpretability undermines their credibility in numerical forecasting applications. Thus, the interpretability of deep learning models is critical to ensuring their predictions align with physical laws (Toms et al., 2020).

For the practical operational implementation of hybrid models, two critical challenges must be addressed, coupling programming issues and model integration interruption. This study focuses on developing feasible solutions to these two key problems from an operational perspective. We adopts data-driven approaches to model the radiation processes within the Global Forecasting System of China Meteorological Administration (CMA-GFS). Guided by the operational mechanism of the radiation process, we first screened the input and output physical quantities for the deep learning model and constructed a well-designed dataset. This step aims to enable the deep learning model to effectively capture and learn the inherent nonlinear characteristics of radiation processes. Subsequently, we analyzed and summarized the key data features of the dataset, which provides critical support for developing effective deep learning parameterization schemes and facilitating physical interpretation of the model. In accordance with the characteristics and mechanisms of coupled operation, a deep learning parameterization scheme was designed and integrated into the host model. A Libtorch-based library is used for real-time operations, which compiles the machine learning emulator into a callable shared library. This library is then linked in the compilation and linking phase to generate the executable program. To enhance the stability of the hybrid model, we leverage prior knowledge to reduce the randomness of the machine learning model, and adopt strategies such as constraint imposition and empirical feedback to ensure the stable operation of the hybrid model. This study also presents the performance of the online hybrid model in real operational scenarios.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

2 RRTMG – original radiation scheme in CMA-GFS

Radiative transfer refers to the transport of energy via electromagnetic waves through a gas medium. The Line-by-Line Radiative Transfer Model (LBLRTM, Clough et al., 1992) serves as the foundational code for the development of all radiation related models. LBLRTM calculates the absorption and emission of radiation by gaseous molecules in the atmosphere, while accounting for the effects of thousands of individual absorption lines. Climate and weather prediction models require fast calculations of radiative flux and heating rates to achieve accurate simulations (Price et al., 2014). The Rapid Radiative Transfer Model (RRTM) has been developed for both the longwave and shortwave regions as reference broadband radiative transfer models that closely reproduce line-by-line results (Mlawer et al., 1997). Absorption coefficients for the primary and minor molecular species required for the correlated k distribution method used by RRTM have been obtained from LBLRTM. Molecular absorbers included in RRTM are water vapor, carbon dioxide, ozone, methane, nitrous oxide, oxygen, nitrogen and the halocarbons in the longwave and water vapor, carbon dioxide, ozone, methane and oxygen in the shortwave.

RRTM retains the highest accuracy relative to line-by-line results for single-column calculations. It has since been modified to develop RRTMG (Iacono et al., 2008), a version optimized to deliver higher computational efficiency with minimal accuracy loss, specifically for general circulation model applications. While RRTMG shares the same fundamental physics and absorption coefficients as RRTM, it incorporates several modifications to improve computational efficiency, and to represent subgrid-scale cloud variability. In particular, the total number of quadrature points (g points) used to calculate radiances in the longwave has been reduced from the standard 256 in RRTM_LW (with 16 g points in each of the 16 spectral bands) to 140 in RRTMG_LW (with the number of g points in each spectral band varying from 2 to 16 depending on the absorption in each band). In the shortwave, the number of g points has been reduced from the standard 224 in RRTM_SW (with 16 in each of the 14 spectral bands) to a total of 112 in RRTMG_SW. RRTM computes multiple scattering with the discrete ordinates algorithm, while RRTMG uses a two-stream algorithm (Oreopoulos and Barker, 1999). Subgrid-scale cloud variability is not considered in RRTM. To improve the realism of cloud-radiation interactions, RRTMG incorporates the Monte-Carlo independent column approximation (McICA) method (Barker et al., 2008).

RRTMG continues to be extensively validated with measured atmospheric spectra from the sub-millimeter to the ultraviolet. Its applications include the IFS system of the ECMWF; the Community Earth System Model and the WRF regional forecast model at NCAR; the CMA-GFS model at China Meteorological Administration Earth System Modeling and Prediction Centre. Although RRTMG is a fast algorithm for representing atmospheric radiative transfer, it remains the most computationally expensive component among the various physical process parameterizations. If radiative transfer impacts were computed for every grid point at all time steps, the associated computational cost would typically be equivalent to that of the model's remaining dynamical and other physical parameterizations combined. To reduce computational time, the radiation scheme is usually invoked only at large temporal intervals and on a coarsened grid. Subsequently, the radiative effects are interpolated to the finer model grid using a cubic interpolation scheme (Morcrette et al., 2008). Specifically, in the

CMA-GFS model, RRTMG is called once per hour and computed on a reduced grid aligned with latitude parallels. Additionally, a load balancing strategy is implemented, as shortwave radiation calculations are only required for half of the globe. Nevertheless, RRTMG still accounts for the largest share of computational time among all physical processes in the model. To address this remaining inefficiency, this study aims to further improve computational efficiency by simulating radiation processes using deep neural networks.

3 Dataset

Following the data interaction mechanism between RRTMG and the host model, the input and output variables generated during the operation of the RRTMG scheme are adopted as the training and testing data. For a given atmospheric column, RRTMG takes atmospheric state variables provided by the CMA-GFS model as inputs and computes radiative fluxes and heating rates as outputs. Table 1 presents the basic information (e.g., names and dimensions) of the input and output physical variables for longwave and shortwave radiation within a single vertical grid column, respectively. The input variables include pressure, temperature, gas mixing ratios, and cloud-related parameters. Variables with vertical height layers are vectors of length 88 or 89, corresponding to the number of model layers. Cloud-related parameters are associated with different absorption bands and have one additional dimension compared to conventional one-dimensional vectors. In contrast, physical variables without height information are scalars. The output variables comprise all-sky upward/downward radiative fluxes, all-sky heating rates, as well as clear-sky upward/downward radiative fluxes and clear-sky heating rates. It is worth noting that to reduce model complexity and account for the influence of different variables on the radiative fluxes and heating rates, we selected a subset of variables from RRTMG as the inputs and outputs for the neural network.

For shortwave radiation, an input-output dataset covering the entire year of 2022 was constructed. It has a spatial resolution of 25 km, an output interval of 25 hours, and was generated by consecutively running 5-day forecasts with sequentially incremented dates. During model training, the dataset was randomly sampled to form a 4.9 TB training set and a roughly 500 GB validation set to fully capture the diverse states of shortwave radiation in real numerical simulations. An equivalent input-output dataset was also constructed for longwave radiation with identical spatial resolution (25 km) and output interval (25 hours). It was generated using the same method by consecutively running 5-day forecasts with sequentially incremented dates to cover the entire year of 2022. For model training, this dataset was randomly sampled to create a 4.2 TB training set and an approximately 450 GB validation set.

Radiation schemes are typically one-dimensional, meaning that each atmospheric column is processed independently. Each sample uses a single vertical grid column as the unit and includes information on various types of input and output physical variables. These samples cover a range of physical variables with distinct properties, which introduces heterogeneity into the dataset. When working with heterogeneous data, the first step is to normalize the data. Normalizing input data helps the machine learning model more effectively extract meaningful features, while normalizing output data prevents the optimizer from over-optimizing physical variables with large numerical scales during training. In this study,

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

standard normalization was applied to the original radiation data to reduce differences in the orders of magnitude across
165 different physical variables.

Table 1: Input and output variables for the networks and training phase.

	Inputs		
Variables	Meaning and dimension	Longwave	Shortwave
play	Layer pressures (hPa, mb) (ncol, nlay)	√	√
plev	Interface pressures (hPa, mb) (ncol, nlay+1)	√	√
tlay	Layer temperatures (K) (ncol, nlay)	√	√
tlev	Interface temperatures (K) (ncol, nlay+1)	√	√
tsfc	Surface temperatures (K) (ncol)	√	√
h2ovmr	H2O volume mixing ratio (ncol, nlay)	√	√
o3vmr	O3 volume mixing ratio (ncol, nlay)	√	√
co2vmr	CO2 volume mixing ratio (ncol, nlay)	√	√
ch4vmr	Mathane volume mixing ratio (ncol, nlay)	√	√
n2ovmr	Nitrous oxide volume mixing ratio(ncol, nlay)	√	√
o2vmr	Oxygen volume mixing ratio (ncol, nlay)	√	√
asdir	UV/vis surface albedo direct rad (ncol)		√
aldir	Near-IR surface albedo direct rad (ncol)		√
asdif	UV/vis surface albedo: diffuse rad (ncol)		√
aldif	Near-IR surface albedo: diffuse rad (ncol)		√
cfc11vmr	CFC11 volume mixing ratio (ncol, naly)	√	
cfc12vmr	CFC12 volume mixing ratio (ncol, nlay)	√	
cfc22vmr	CFC22 volume mixing ratio (ncol, nlay)	√	
ccl4vmr	CCL4 volume mixing ratio (ncol, nlay)	√	
tauaer	Aerosol optical depth at mid-point of LW spectral bands (ncol, nlay, nbndlw)	√	
ecaer	Aerosol optical depth at 0.55 micron(ncol, nlay, naerec)		√
taucmcl	In-cloud optical depth (ngptlw, ncol, nlay)	√	
reicmcl	Cloud ice particle effective size (ncol, nlay)	√	
relqmcl	Cloud water drop effective radius (ncol, nlay)	√	

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

ciwpmcl	In-cloud ice water path (g/m2)(ngptsw, ncol, nlay)		√
clwpmcl	In-cloud liquid water path (g/m2)(ngptsw, ncol, nlay)		√
reicmcl	Cloud ice effective radius (microns) (ncol, nlay)		√
relqmcl	Cloud water drop effective radius (ncol, nlay)		√
emis	Surface emissivity (ncol, nbndlw)	√	
coszen	Cosine of solar zenith angle (ncol)		√
dyofyr	Day of the year		√

Outputs			
Variables	Meaning and dimensions	Longwave	Shortwave
uflx	Total sky longwave upward flux (W/m2) (ncol, nlay+1)	√	
dflx	Total sky longwave downward flux (W/m2) (ncol, nlay+1)	√	
hr	Total sky longwave radiative heating rate (K/d) (ncol, nlay)	√	
uflxc	Clear sky longwave upward flux (W/m2) (ncol, nlay+1)	√	
dflxc	Clear sky longwave downward flux (W/m2) (ncol, nlay+1)	√	
hrc	Clear sky longwave radiative heating rate (K/d) (ncol, nlay)	√	
swuflx	Total sky shortwave upward flux (W/m2) (ncol, nlay+1)		√
swdflx	Total sky shortwave downward flux (W/m2) (ncol, nlay+1)		√
swhr	Total sky shortwave radiative heating rate (K/d) (ncol, nlay)		√
swuflxc	Clear sky shortwave upward flux (W/m2) (ncol, nlay+1)		√
swdflxc	Clear sky shortwave downward flux (W/m2) (ncol, nlay+1)		√
swhrc	Clear sky shortwave radiative heating rate (K/d) (ncol, nlay)		√

4 Machine learning model and training

In this study, we use a convolutional neural network (CNN) as the base model and incorporate residual modules (He et al., 2016). It allows the deep network to effectively learn complex nonlinear information embedded in the data. Input and output variables within a single vertical grid column were stacked separately. By merging the different absorption bands of cloud-related parameters, a multi-channel one-dimensional vector was used as input and a 6-channel one-dimensional vector was used as output. Since some input variables are defined on full model layers and others on half layers, the data were padded to

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

ensure a uniform 89-layer structure. A deep learning parameterization scheme was constructed using RseCNN to predict the radiative fluxes and heating rates within each vertical grid column. This is consistent with the operational principle of the RRTMG scheme. The model incorporates three residual modules, each containing two convolutional layers. Additional fully connected layers were added to capture more global information of the single column, as illustrated in Fig. 1. Other neural networks were also tested, including fully connected neural networks and convolutional block attention networks (CBAMs, Woo et al., 2018). However, when the number of parameters was comparable to that of ResCNN, the fully connected model exhibited relatively poor representational capability. Despite having more parameters, the CBAM model did not lead to a noticeable improvement in prediction accuracy.

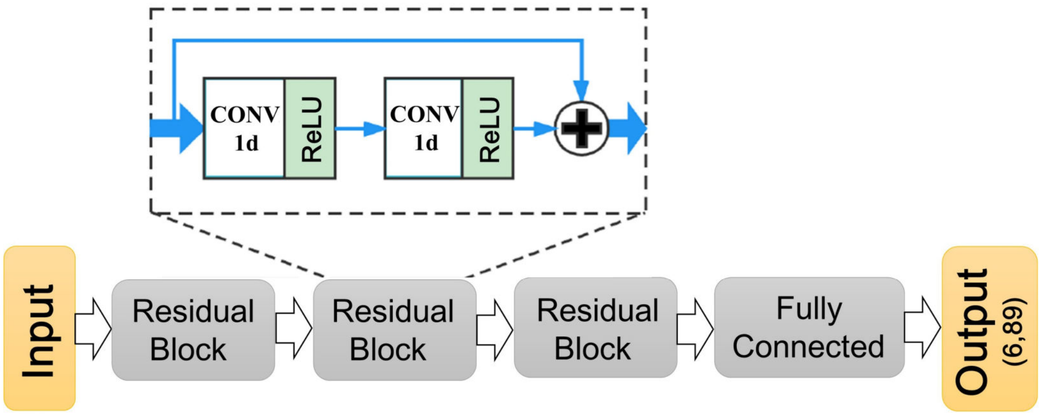

Figure 1: Architecture of the longwave radiation machine learning model.

Model training was conducted using P100 GPUs and based on the PyTorch framework. Given the 'small model, large dataset' configuration adopted for this task, the training process was implemented via Distributed Data Parallelism, which is a built-in distributed data-parallel training module in PyTorch. Taking shortwave radiation as an example, for a 4.9TB training set and a 500GB validation set, training was performed using 8 P100 GPUs, lasting 68 hours and completing a total 80 epochs. The ReLU function was used as the activation function, the initial learning rate was set to 0.005, with cosine annealing adopted as the learning rate adjustment strategy, and the SGD optimizer was employed.

First we conducted offline tests on the radiation machine learning (ML) emulator. As shown in Fig. 2 and Fig. 3, the left panel compares the six variables predicted by the neural network with the reference values output by RRTMG, while the right panel presents the mean absolute error (MAE) of these six variables and their biases across different percentiles. Across both longwave and shortwave radiative quantities, the proposed neural network achieves highly accurate predictions when compared to the benchmark RRTMG scheme. For upward and downward longwave fluxes, the predicted profiles nearly overlap with the truth across all model levels, demonstrating excellent agreement in both magnitude and vertical structure. Minor discrepancies are observed near the model top and lower atmosphere for the longwave heating rate. The mean

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

absolute error remains below 0.005 K/day across all levels, with the 25% and 5% percentile intervals tightly constrained around zero, confirming stable and low-bias performance under both clear-sky and total-sky conditions. For shortwave radiative quantities, the predicted upward and downward shortwave fluxes closely track the reference profiles. The shortwave heating rate exhibits near-perfect alignment with the truth, yet small mismatches appear at the model top where the heating rate magnitude peaks. The MAE for shortwave fluxes stays below 0.01 W/m² across most levels, and for the shortwave heating rate, the MAE remains below 0.015 K/day, with percentile intervals that remain narrow even at the model top where heating rate magnitudes peak. The vertical structure of radiative heating rates and fluxes is faithfully reproduced, with no systematic biases evident in the error distributions. These results confirm that the neural network emulator can reliably replicate the complex radiative transfer processes modeled by RRTMG, making it suitable for integration into numerical weather prediction models as a computationally efficient alternative.

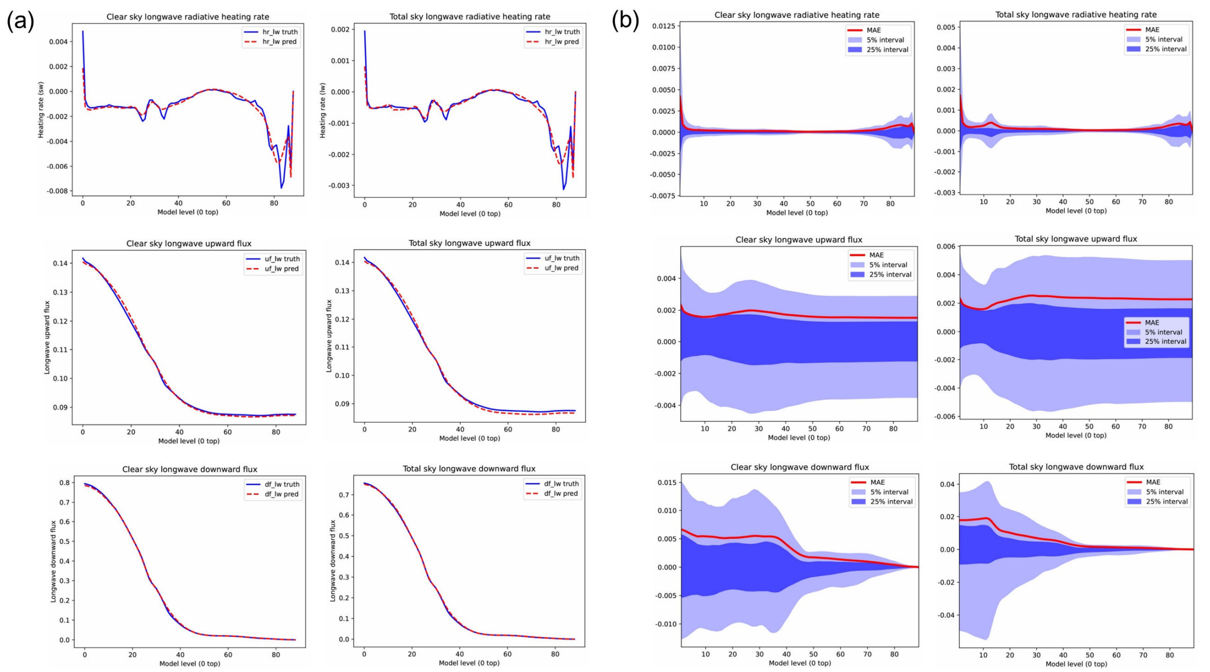

Figure 2: Offline test results of the longwave radiation, (a) comparisons between the predicted variables of the neural network and the reference values, (b) MAE and 5th/25th percentile confidence intervals.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

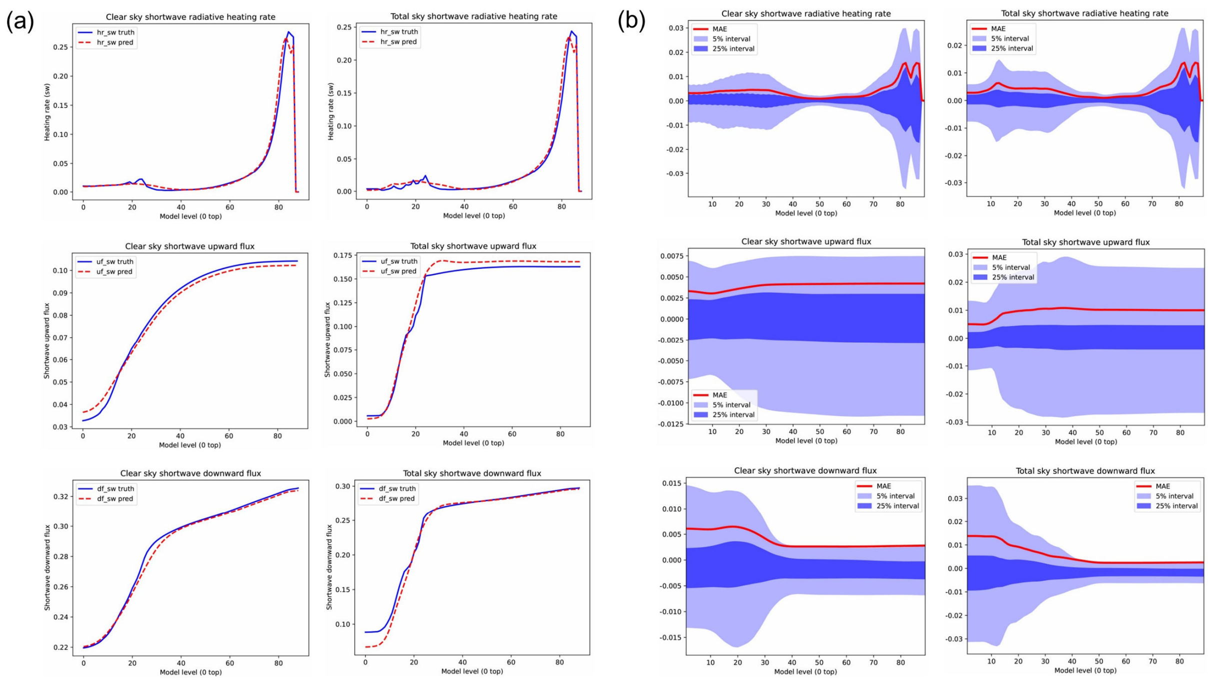

Figure 3: Offline test results of the shortwave radiation, (a) comparisons between the predicted variables of the neural network and the reference values, (b) MAE and 5th/25th percentile confidence intervals.

5 Coupling the ML emulator and CMA-GFS

Deep learning parameterization schemes are mainly trained using Python-based machine learning frameworks. Python is a dynamically typed language, which means deep learning models cannot be directly invoked as functions by numerical models developed in Fortran, a statically typed language.

To couple the ML emulator with traditional numerical models, one approach is to hardcode deep learning models into Fortran code. However, this method has low development efficiency. Any modification to the deep learning model requires corresponding changes to the associated code. To address this, researchers have developed an interface called the fortran-keras deep learning bridge (Otte et al., 2020). This interface enables Fortran programs to import pre-training deep learning parameters (i.e., weights and biases) from external sources. These parameters are stored in TXT files and invoked when the deep learning model performs inference. Notably, the interface can be deployed in numerical models with relative flexibility, as it does not require modifying the deep learning model's network structure. Nevertheless, this interface primarily supports models based on fully connected layers and cannot accommodate many state-of-the-art network architectures, such as diffusion models or Transformer models. Another widely used coupling method splits the Fortran-based host model and Python-based deep learning components into two separate process sets. It enables data transmission via Inter-Process Communication (IPC) while achieving synchronous control during coupled operations (Wang et al., 2022). While this

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

coupling method is ideal for online learning and the initial debugging of model coupling, communication between Fortran and Python processes introduces extra computational latency.

The Fortran Torch Adapter (FTA) is a versatile tool built on LibTorch (Mu et al., 2023), with a design philosophy similar to that of the ECMWF coupling library Infero (https://infero.readthedocs.io/en/latest/index.html). This tool supports converting deep learning models under the PyTorch framework into a static binary format and provides a C/C++ interface. Subsequently, Fortran-C/C++ mixed programming techniques are used to deploy the ML emulator in numerical models.

Integrating ML emulator into the host model involves frequent calls to identical small-scale networks for fast inference and frequent communication. This scenario is highly suited to the LibTorch-based coupling method for real-time operation. We used a LibTorch-based compilation library tool and extended its functionality. Acting as an intermediate layer, the compilation library tool offers Fortran users a simple, transparent interface for operating PyTorch models, while hiding the underlying complexity of connecting to C++ interfaces via language interoperability. We only need to invoke these interfaces like regular Fortran library functions. The compilation library tool automatically handles all trivial inconsistencies between Fortran and C++, such as data types and array memory models. This tool also supports customization based on user-specific needs, with customizable parameters including system environment configurations (e.g., the compiler used and PyTorch version) and usage options (e.g., the dimensions and types of input/output data for the ML emulator, whether to utilize GPU resources, and the ML emulator's storage location). This coupling method provides high flexibility. However, it still faces certain code compatibility issues in mixed programming. Additionally, since LibTorch is based on C/C++ programming standards, debugging and stability testing during coupled operations remain less flexible and efficient.

Subsequently, by writing code at the locations where network activation is required, we can enable calls to the PyTorch network trained offline. In this setup, the numerical model and the ML emulator perform computations within the same computing process, eliminating the need for additional communication. This makes such a coupling approach ideal for practical operational applications. The mechanism of the compilation library tools is illustrated in Fig. 4.

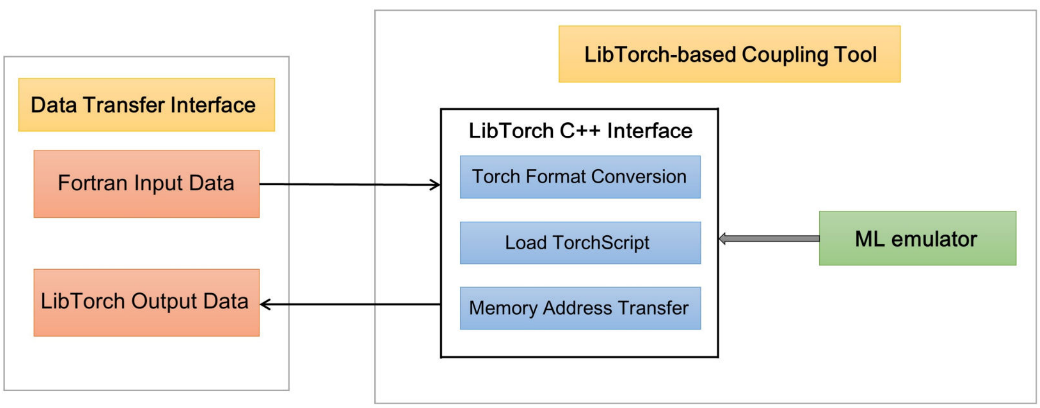

Figure 4: The mechanism of the LibTorch-based coupling tool.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

6 Enhance the stability of the hybrid model

The radiation ML emulator is coupled with the CMA-GFS model via a compiled library tool, resulting in a hybrid model. Once the network is sufficiently trained, the deep learning model can approximate the accurate solutions of radiative processes generated by RRTMG. Due to the application of standard normalization, the output values of the deep learning model are constrained within the interval 0 and 1. These values are then multiplied by a normalization coefficient to generate the final output, which is transmitted to the host model. For radiation processes, although the training data were generated at a 25 km resolution, the hybrid model can still operate at a 12.5 km resolution and support 10-day integral forecasts required for operational purposes.

For the hybrid model, when both the ML emulator and the host model run on CPU nodes, each process handles an equal number of samples. As shown in Table 2, the computational efficiency of the ML emulator is improved by nearly eightfold on average compared with the original RRTMG. Furthermore, we compare the time consumption of the ML emulator and the original RRTMG module on different parallel computing scales. It can be seen that RRTMG has excellent scalability. The ML emulator also exhibits good scalability, though slightly inferior to that of the physical model. When the ML emulator is executed on a GPU node, its inference speed is inherently enhanced. However, the data transmission latency must be considered.

Table 2: Comarison of computation time: ML emulator in the hybrid model vs. RRTMG in the original model.

Number of CPU cores	1920	3840	7680	11520	15360
Radiation ML emulator	0.204s	0.093s	0.053s	0.037s	0.028s
RRTMG	1.441s	0.715s	0.342s	0.228s	0.177s

We conducted 50 experiments using the initial version of the hybrid model, with ten-day forecasts initiated at randomly selected times in 2023. Among these experiments, 18% experienced interruptions. When such interruptions occurred, we traced and identified the coupling breakpoints between the ML emulator and the host model. Most interruptions were found to originate from a critical issue: the uncertainty caused by extrapolating extreme data beyond the training set poses a major challenge to the emulator. This uncertainty may lead to the breakdown of the entire model during long-term simulations, particularly in extreme weather conditions or edge scenarios such as typhoons and low-temperature environments in high-latitude regions. The long-term stable coupled operation of the deep learning model and the host model is a prerequisite for transitioning deep learning parameterization schemes form research to practical operational applications. We analyse the mechanisms underlying numerical instability in the hybrid model. Let $\mathrm{x_t}$ denote the atmospheric state vector at time t. The single-step integration of a conventional numerical model can be written as:

$$\mathrm{x_{t+1}} = M(\mathrm{x_t}) + P(\mathrm{x_t}), \tag{1}$$

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

where M represents the dynamical core and other physical processes, and P denotes the physical radiation parameterization. In the hybrid model, we replace P with a neural network emulator f_θ. The forecast trajectory of the hybrid model is then given by:

$$\tilde{\mathrm{x}}_{\mathrm{t}+1} = M(\tilde{\mathrm{x}}_{\mathrm{t}}) + f_\theta(\tilde{\mathrm{x}}_{\mathrm{t}}). \quad (2)$$

Define the approximation error of the neural network as $\epsilon(x) = f_\theta(x) - P(x)$. Within the range of typical weather regimes covered by the training dataset, a well-trained neural network closely approximates the reference radiation scheme, yielding small errors $\|\epsilon(x)\| \leq \delta$, where δ is a small constant. In contrast, for out-of-distribution conditions such as tropical cyclones and other extreme weather scenarios, the neural network exhibits uncontrolled extrapolation behaviour due to insufficient training data. This leads to severe and unphysical errors, such that $\|\epsilon(x)\| \geq \delta$.

Let $e_t = \tilde{x}_t - x_t$ denote the system state error of the hybrid model at time t. The error at the next time step evolves as:

$$e_{\mathrm{t}+1} = [M(\tilde{\mathrm{x}}_{\mathrm{t}}) + f_\theta(\tilde{\mathrm{x}}_{\mathrm{t}})] - [M(\mathrm{x}_{\mathrm{t}}) + P(\mathrm{x}_{\mathrm{t}})]. \quad (3)$$

To isolate individual error contributions, we add and subtract $P(\tilde{\mathrm{x}}_{\mathrm{t}})$:

$$e_{\mathrm{t}+1} = [M(\tilde{\mathrm{x}}_{\mathrm{t}}) - M(\mathrm{x}_{\mathrm{t}})] + [P(\tilde{\mathrm{x}}_{\mathrm{t}}) - P(\mathrm{x}_{\mathrm{t}})] + [f_\theta(\tilde{\mathrm{x}}_{\mathrm{t}}) - P(\tilde{\mathrm{x}}_{\mathrm{t}})]. \quad (4)$$

Assuming differentiability of M and P in the neighbourhood of the current state, we perform a first-order Taylor expansion and neglect higher-order terms. Let J_M and J_P denote the Jacobian matrices of M and P respectively. The error propagation can then be approximately as:

$$e_{\mathrm{t}+1} \approx J_M(\mathrm{x}_{\mathrm{t}})e_{\mathrm{t}} + J_P(\mathrm{x}_{\mathrm{t}})e_{\mathrm{t}} + \epsilon(\tilde{\mathrm{x}}_{\mathrm{t}}), \quad (5)$$

$$e_{\mathrm{t}+1} \approx \big(J_M(\mathrm{x}_{\mathrm{t}}) + J_P(\mathrm{x}_{\mathrm{t}})\big)e_{\mathrm{t}} + \epsilon(\tilde{\mathrm{x}}_{\mathrm{t}}). \quad (6)$$

Taking norms to quantify error growth yields:

$$e_{\mathrm{t}+1} = \|J_M + J_P\| \cdot \|e_{\mathrm{t}}\| + \|\epsilon(\tilde{\mathrm{x}}_{\mathrm{t}})\|, \quad (7)$$

where $\mathrm{L} = \|J_M + J_P\|$ is defined as the local Lipschitz constant or error amplification factor of the coupled system. Model breakdown arises from two superimposed mechanisms. The first is explosive input perturbations. As the model integrates into extreme weather regimes, the neural network generates large unphysical disturbances that rapidly inflate $\|\epsilon(\tilde{\mathrm{x}}_{\mathrm{t}})\|$. The second is insufficient physical regularization. Real physical processes inherently incorporate dissipative mechanisms (e.g. radiative cooling that damps extreme temperature anomalies). When the neural networks lacks such physically consistent dissipation, its local Lipschitz constant can becomes significantly greater than unity. Under sustained injection of large out-of-distribution errors, the error sequence $\|e_t\|$ grows exponentially, triggering numerical instability and eventually leading to floating-point overflow. This manifests as catastrophic model failure during long-term simulations.

To address the instability issue, we considered the following solutions. First, leverage prior knowledge and the physical process logic of numerical models to reduce the blindness of machine learning model. The heating rate predicted by the ML emulator shows a relatively large error. The required heating rate can be calculated using the following relationship:

$$Heating\_rate[i] = -\frac{g}{c_p} \cdot \frac{flux\_dn[i] + flux\_up[i] - flux\_dn[i+1] - flux\_up[i+1]}{hl\_pressure[i] - hl\_pressure[i+1]}, \quad (8)$$

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

where $Heating\_rate[i]$ denotes the atmospheric heating rate at layer i, g represents gravitational acceleration, c_p stands for the specific heating capacity of air at constant pressure, $flux\_dn[i]$ is the downward radiative flux at layer i, $flux\_up[i]$ is the upward radiative flux at layer i, and $hl\_pressure[i]$ refers to the atmospheric pressure at layer i. Second, constrain the output range of the ML emulator by setting reasonable upper and lower bounds for output variables based on their physical significance. We specify reasonable upper and lower bounds $[f_{min}, f_{max}]$, given the raw output of the neural network f_{pred}, the final variable passed to the model is $\hat{f} = \max\left(\min\left(f_{pred}, f_{max}\right), f_{min}\right)$. This prevents the ML model from producing unphysical predictions caused by input noise or extreme values. Third, we optimize the training dataset by collecting extreme samples that triggered interruptions in previous experiments to construct a new refined dataset, which is subsequently merged with the original dataset. This strategy broadens the feature distribution of the training set and suppresses unconstrained extrapolation by the model when encountering complex meteorological conditions. In addition, random noise is injected into the training data to improve the model's robustness to input perturbations. From an optimization perspective, adding random noise yields a smoother loss landscape during neural network weight optimization. This prevents the model from converging to highly fragile local minima and thus maintains stability within the hybrid model. Testing results indicated that the third solution played the most significant role in maintaining the integration stability of the hybrid model. All 50 selected experiments successfully completed the ten-day forecasts without interruptions. This was achieved without modifying the structure of the machine learning models. The hybrid model was subsequently operated in real time once per day, with no integration interruptions occurred over a four-month period.

7 Performance in the operational reforecast experiments

The operational version of the CMA-GFS model runs at a horizontal resolution of 12.5 km. We conducted a two-month operational reforecast experiment on the hybrid model, covering January 2024 and July 2024. The results were compared against those from the operational system. First, we compared the forecast performance of synoptic fields, as shown in Fig. 5. The hybrid model and the operational model showed comparable performance in synoptic field forecasting, with a slight reduction in predictable days observed over the Northern Hemisphere in January.

Next, we verify the precipitation forecasts, with results presented in Fig. 6 to Fig. 9. For the Threat Score (TS), the hybrid model outperformed the operational model in several scenarios. In January, it showed advantages in forecasting light and moderate rain within the first 3 days of lead time. In July, it excelled at predicting heavy and torrential rain within the first 1.5 days. For other lead times and rainfall grades, the hybrid model's precipitation forecasts were either comparable to or less accurate than those of the operational model. Regarding precipitation forecast bias, in January, it exhibited lower bias than the operational model across all rainfall grades within the first 2 days of lead time. However, this advantage reversed beyond the 2-day mark with its bias becoming higher than that of the operational model. In July, the hybrid model showed a more grade-specific pattern. It had higher bias for light and moderate rain, yet lower bias for heavy and torrential rain regardless of lead time. For accumulated precipitation assessment, the hybrid model barely changed the forecast of daily

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

accumulated precipitation below 10 mm compared with the operational model. For daily accumulated precipitation above 10 mm, its precipitation distribution was more consistent with observations, with a slight increase in the forecast of precipitation extremes.

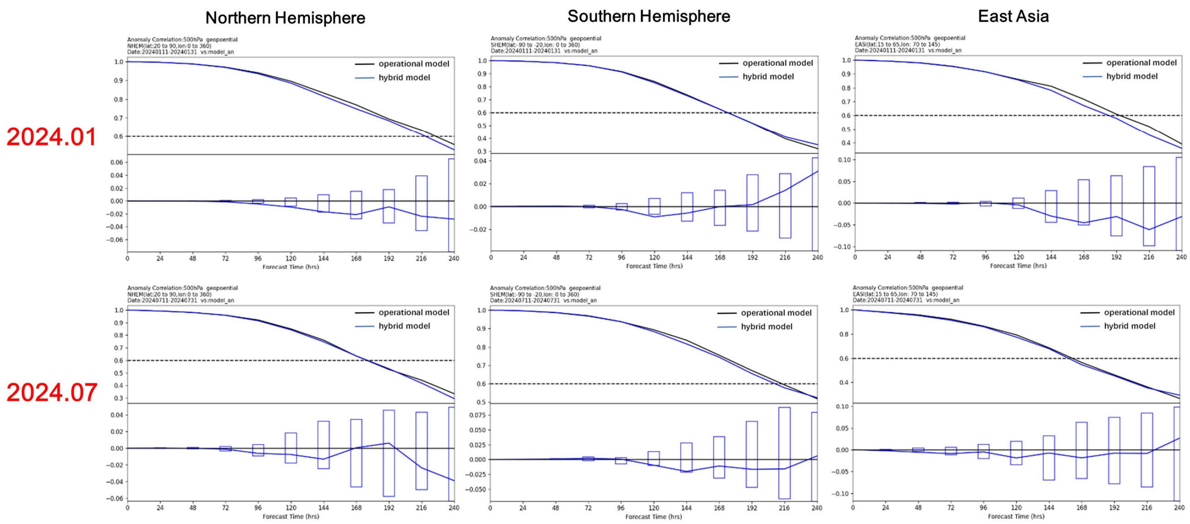

Figure 5: Synoptic fields comparisons between the hybrid and operational models in January and July.

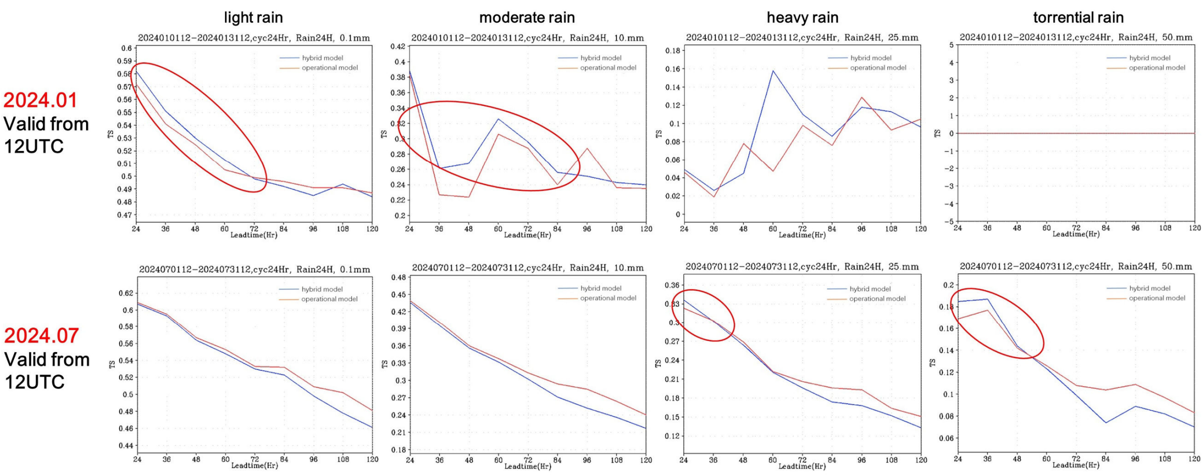

Figure 6: Comparisons of Threat Scores for precipitation forecasts: hybrid model vs. operational model.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

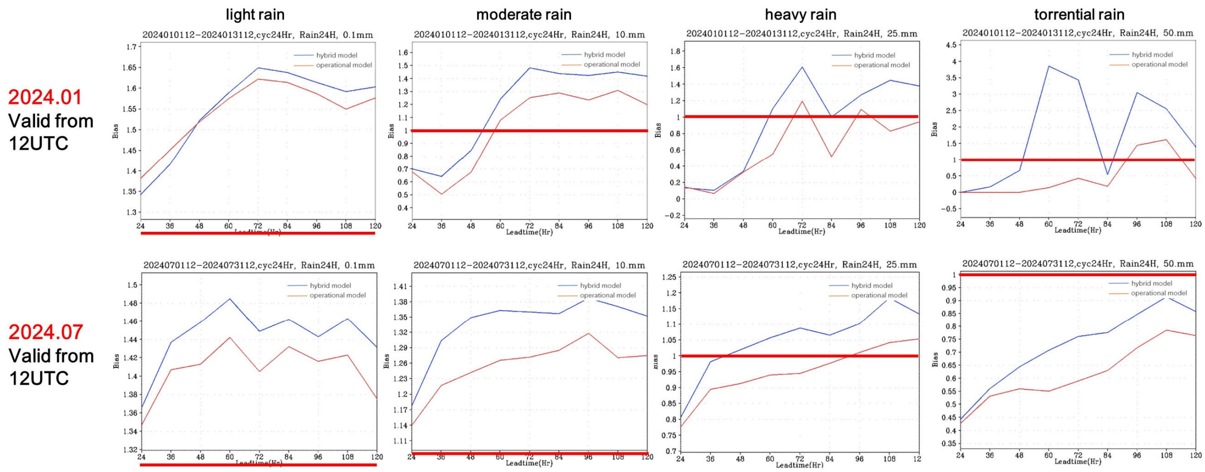

Figure 7: Comparisons of precipitation forecast biases: hybrid model vs. operational model.

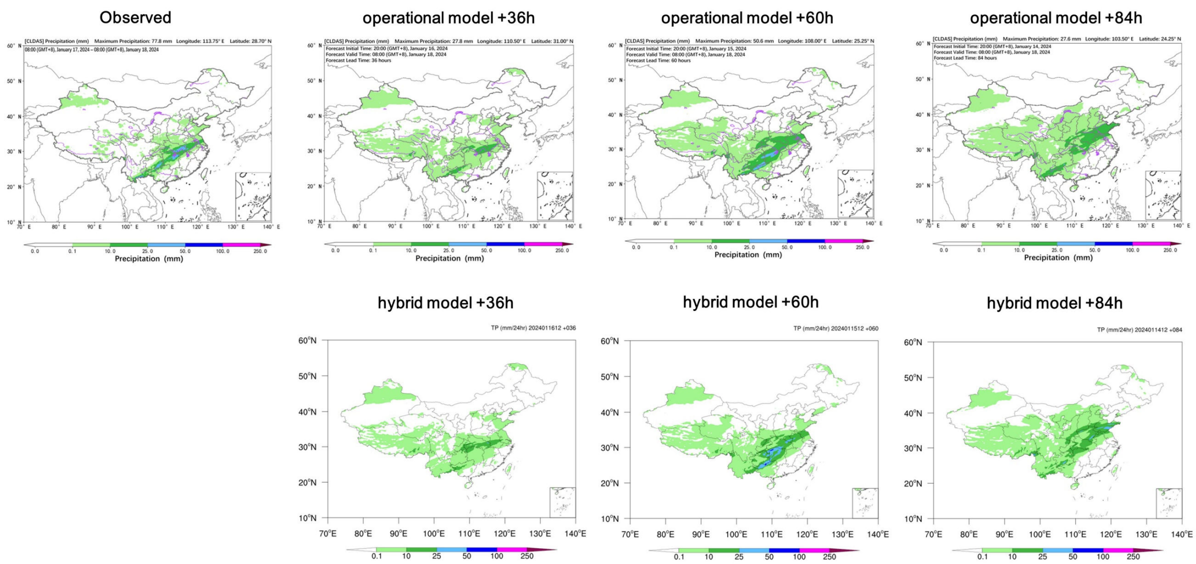

Figure 8: 24-hour accumulated precipitation from 08:00 on January 17 to 08:00 on January 18, 2024.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

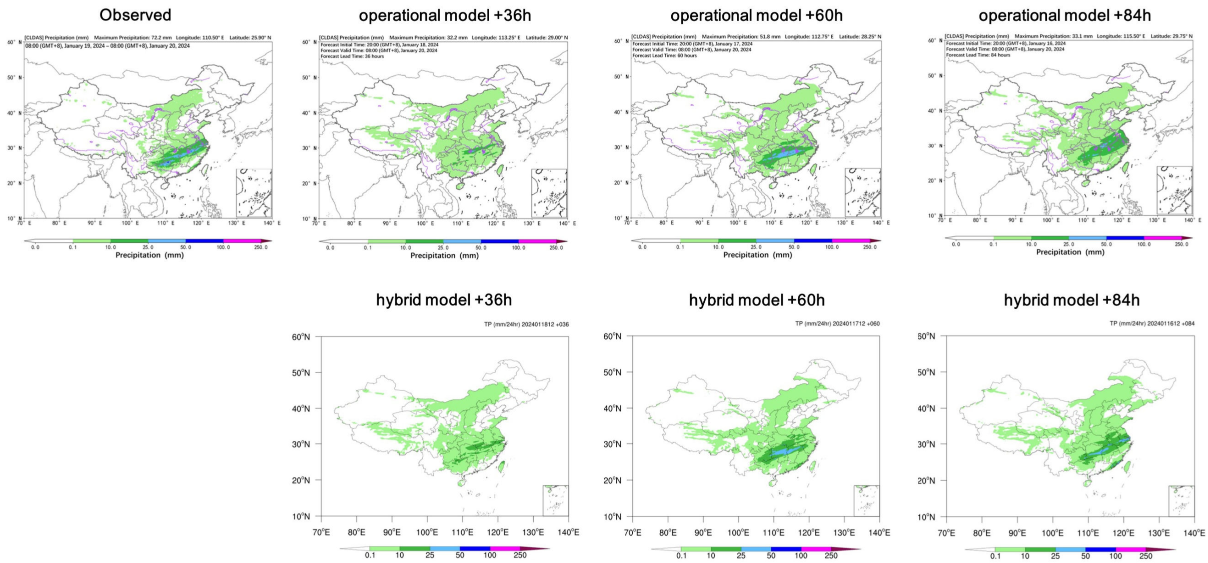

Figure 9: 24-hour accumulated precipitation from 08:00 on January 19 to 08:00 on January 20, 2024.

Verification results for 2-meter temperature forecasts are presented in Fig. 10. In January, the absolute error of the hybrid `model's 2-meter temperature forecasts was generally comparable to that of the operational model. In July, the hybrid model outperformed the operational model within the first 1.5 days of lead time, but the operational model became more accurate after the second day. Verification results for 10-meter wind speed forecasts are shown in Fig. 11. In January, the hybrid model exhibited higher absolute error in 10-meter wind speed forecasts than the operational model. Additionally, its relative error for zonal wind was significantly higher within a 3-day lead time. In July, the hybrid model's absolute error in 10-meter wind speed forecasts also exceeded that of the operational model, with notably higher relative error for meridional wind observed across all validity periods.

A severe cold wave event occurred from January 19 to 20, 2024. For the operational model, its forecast of the temperature change area was consistent with observations but slightly weaker in intensity, particularly failing to capture the temperature change center intensity over central Inner Mongolia. For the hybrid model, the 24-hour forecast was largely consistent with the operational model. However, its 48-hour and 72-hour forecasts showed a slightly stronger intensity of the temperature change center in central Inner Mongolia than the operational model, as illustrated in Fig. 12. Verification results for a heat wave event are presented in Fig. 13. For the heat wave from July 22 to 23, 2024, the operational model's forecasts of both heat wave range and intensity were weaker than observations. In contrast, the hybrid model's 24-hour forecast of the heat wave range was closer to observations, while its 48–72 hour forecasts indicated a smaller heat wave range in the Xinjiang region compared to the operational model's results.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

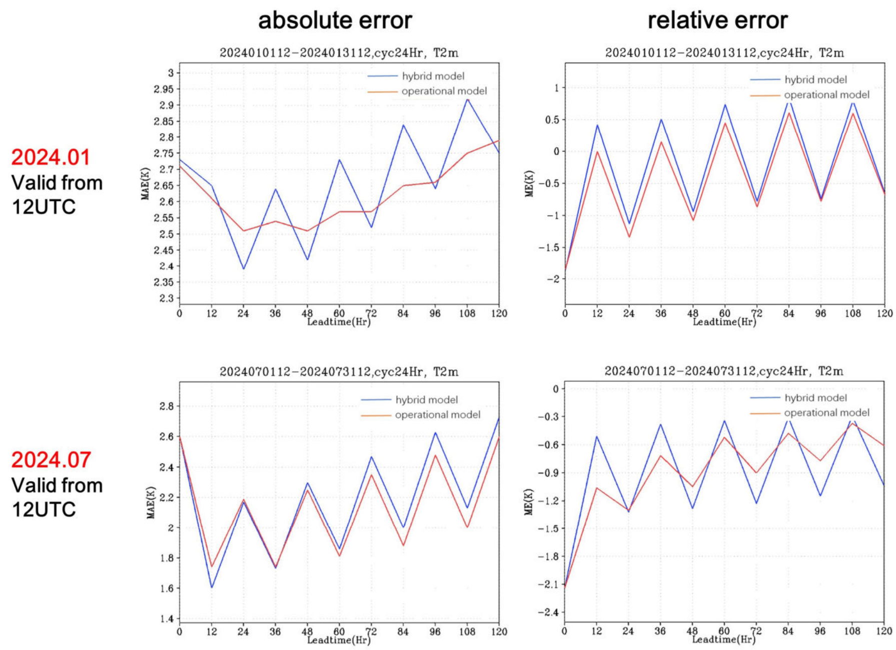

Figure 10: Comparison of 2-meter temperature forecasts: hybrid model vs. operational model.

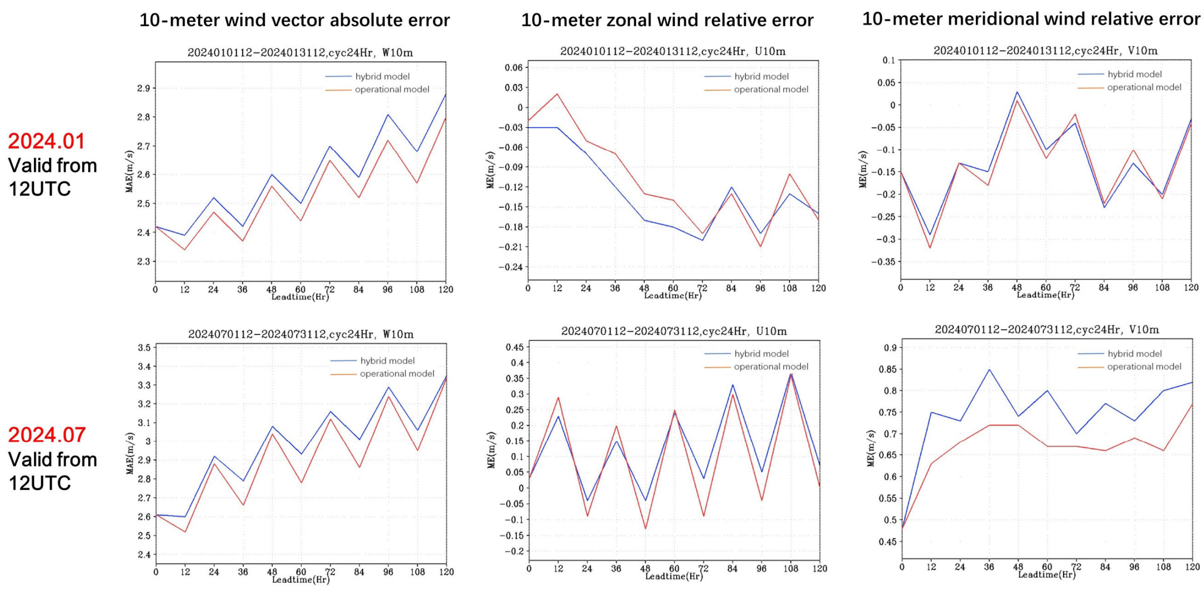

Figure 11: Comparison of 10-meter wind speed forecasts: hybrid model vs. operational model.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

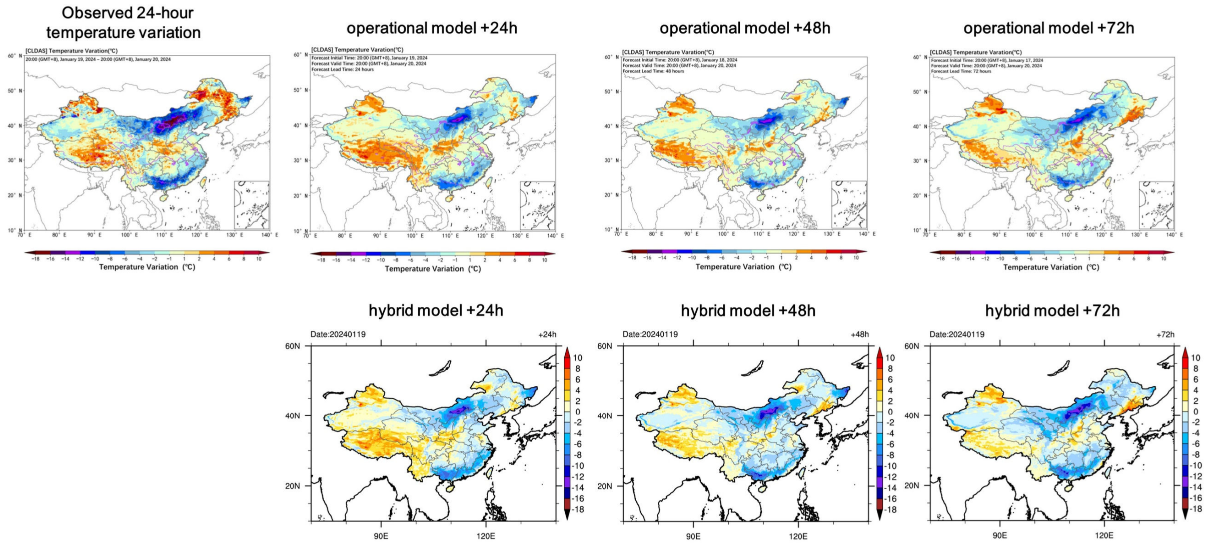

Figure 12: Verification of cold wave forecast performance (January 2024).

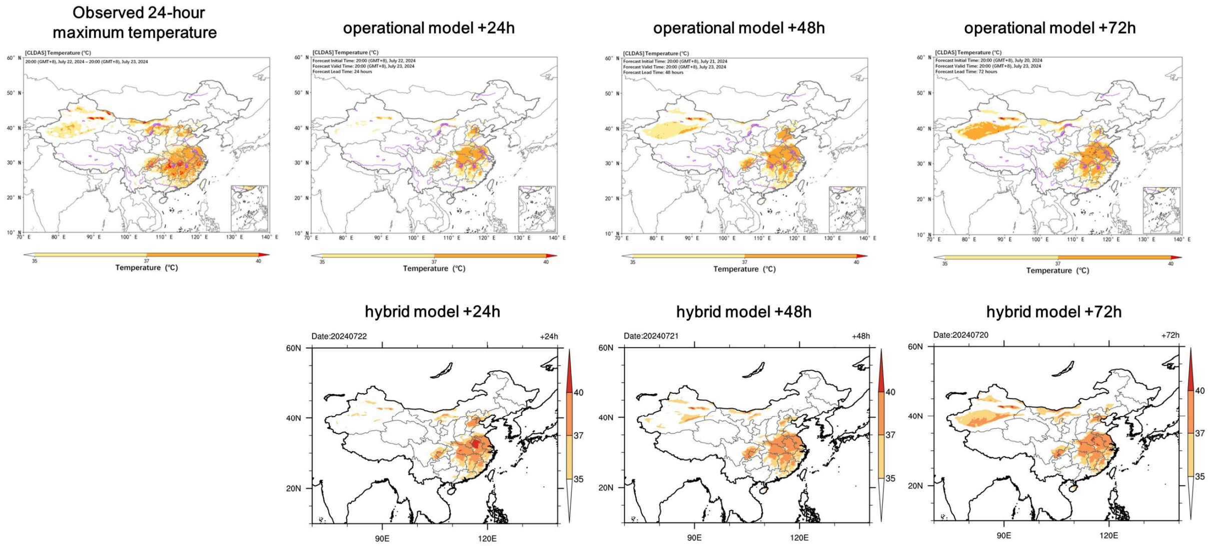

Figure 13: Verification of heat wave forecast performance (July 2024).

Typhoon Kaemi formed over the ocean east of the Philippines on the afternoon of July 20, 2024. It then gradually developed and intensified, becoming the first super typhoon of 2024 with its peak intensity reached around the afternoon of July 22. Typhoon Kaemi was characterized by a massive cloud system, strong intensity, and wide-ranging impacts. A comparison of forecast results for Typhoon Kaemi's precipitation is presented in Fig. 14(a). Both the hybrid model and the operational model underestimated the intensity of typhoon-induced heavy precipitation. Relative to the operational model,

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

the hybrid model showed no significant improvement in precipitation forecasting, though its predicted precipitation intensity was slightly stronger. Additionally, the hybrid model and the operational model exhibited comparable performance in forecasting Typhoon Kaemi's track and intensity, as illustrated in Fig. 14(b).

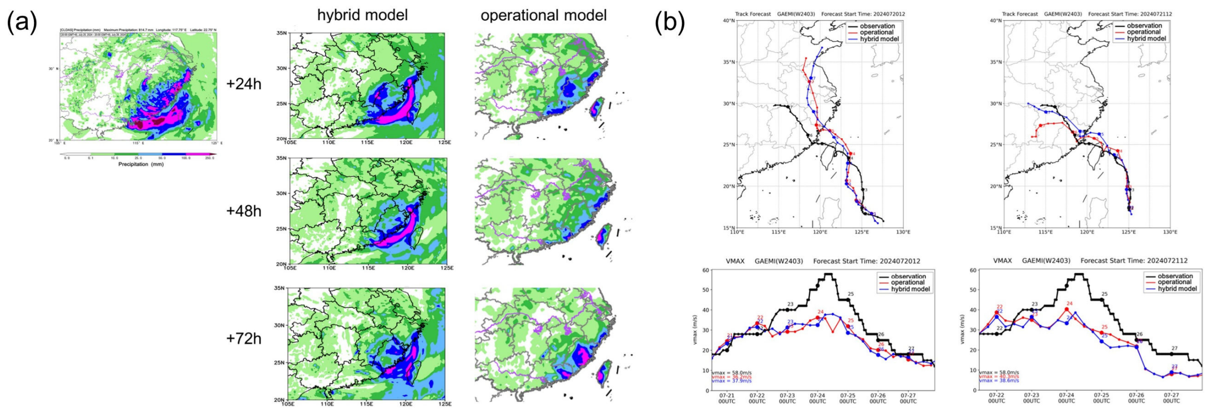

Figure 14: A comparison of Typhoon Kaemi forecasts: (a) precipitation, (b) track and intensity.

Overall, the current hybrid model achieves forecast performance comparable to that of the operational model. However, its performance in forecasting synoptic fields and 10-meter wind speed is slightly diminished — this problem can be addressed by incorporating additional information, such as satellite observation data or LBLRTM physical model. Given that the current deep learning radiation model was initially developed solely to approximate the original RRTMG scheme, the operational reforecast experiments conducted above demonstrate that the predefined objectives have been achieved.

8 Discussion and future work

This study developed a machine learning-based parameterization scheme for the radiation physical processes in the CMA-GFS model. It also proposed exclusive solutions to the problems encountered by the hybrid model under operational configurations. The ML emulator was converted to C++ format using Libtorch to enable subsequent invocation by the host model. Stability issues of the hybrid model were resolved via incorporating physical significance, dataset augmentation and experience replay, facilitating stable online coupling. Additionally, this study verified the transferability of the ML emulator across different model resolutions. Even though the ML emulator performed inference on CPU nodes, its computational speed was eight times faster than that of the original physical scheme. A two-month operational reforecast experiment was conducted on the improved hybrid model, followed by a comprehensive evaluation. Results demonstrated that the hybrid model achieved simulation accuracy comparable to the operational model. A limitation of this approach is that the neural network requires retraining whenever configurational adjustments are made, such as changes to vertical resolution. Future work will focus on further optimizing and refining the neural network model. Using a line-by-line model fully grounded in

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

physical laws to correct the output variables of the training dataset, or integrating observation data. Both strategies aim to improve forecast accuracy.

Code and data availability

The hybrid model based on CMA-GFS v4.2, together with the training codes and CMA-GFS-derived training data, are archived at https://doi.org/10.5281/zenodo.20117982 (Jing, 2026). Due to data volume limitations, only a subset of the training data is publicly available in this repository. Forecast outputs from both the CMA-GFS and hybrid models, initialized at 12:00 UTC on 1 January 2024, are released as one snapshot per day and can be accessed at https://doi.org/10.5281/zenodo.20122347 (Jing, 2026).

Author contributions

Hao Jing designed the overall research framework, implemented the code for neural network training and addressed stability issues, conducted the relevant experiments, figure plotting, and manuscript drafting. Sa Xiao generated the training data and implemented the coupling between the machine learning emulator and the numerical model. The other authors contributed to different aspects of the study, including discussions on data processing, algorithm details and manuscript revision.

Competing interests

The authors declare that they have no competing interests.

Disclaimer

Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements

We would like to express our gratitude to Dr. Siyuan Sun and Dr. Bin Zhao for their help in evaluating the model. We also acknowledge the use of AI tools for English polishing and grammar checks during the preparation of this manuscript.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

Financial support

This work was supported by the National Natural Science Foundation of China (Grant No. U2242210 and No. U2342220).

References

Alves, P., Bovalo, C., Chantry, M., Druilhe, R., Guibert, D., Marradi, L., and Vincent, L.: Representing the 3D cloud radiative effects in ecRad using machine learning, 20th ECMWF workshop on high performance computing in meteorology, 2023.

Arcucci, R., Zhu, J., Hu, S., and Guo, Y. K.: Deep data assimilation: integrating deep learning with data assimilation, Applied Sciences, 11(3), 2021.

Barker, H., Cole, J., Morcrette, J. J., Pincus, R., Raisanen, P., Salzen, K., and Vaillancourt, P. A.: The Monte Carlo Independent Column Approximation: an assessment using several global atmospheric models, Quarterly Journal of the Royal Meteorological Society, 134: 1463-1478, 2008.

Brenowitz, N. and Bretherton, C.: Spatially extended tests of a neural network parametrization trained by coarse-graining, Journal of Advances in Modeling Earth Systems, 11(8): 2728-2744, 2019.

Chantry, M., Hatfield, S., Dueben, P., Polichtchouk, I., and Palmer, T.: Machine learning emulation of gravity wave drag in numerical weather forecasting, Journal of Advances in Modeling Earth Systems, 13.7, 2021.

Chen, J., Zhang, M., Zhang, T., Lin, W., and Xue, W.: Stable simulation of the community atmosphere model using machine-learning physical parameterization trained with experience replay, Journal of Advanceds in Modeling Earth Systems, 17(6), 2025.

Chen, Y., Huang, G., Wang, Y., Tao, W., Tian, Q., Yang, K., Zheng, J., and He, H.: Improving the Heavy Rainfall Forecasting Using a Weighted Deep Learning Model, Frontiers in Environmental Science, 11, 2023.

Chevallier, F., Morcrette, J., Chéruy, F., and Scott, N. A.: Use of a neural-network-based long-wave radiativetransfer scheme in the ecmwf atmospheric model, Quarterly Journal of the Royal Meteorological Society, 126(563): 761-776, 2000.

Clough, S. A., Iacono, M. J., and Moncet, J.: Line-by-line calculations of atmospheric fluxes and cooling rates: Application to water vapour, Journal of Geophysical Research, 97(D14):15,761-15,785, 1992.

Cybenko, G.: Approximation by superpositions of a sigmoidal function, Mathematics of control. Signals and Systems, 2(4): 303-314, 1989.

ECMWF STRATEGY, 2025-2034. https://www.ecmwf.int/en/about/what-we-do/strategy, 2025.

Gao, S., Lu, C., Zhu, J., Li, Y., Liu, Y., Zhao, B., Hu, S., Liu, X., and Lv, J.: Using Machine Learning to Predict Cloud Turbulent Entrainment-Mixing Processes, Journal of Advances in Modeling Earth Systems, 16(8): e2024MS004225, 2024.

Hafner, K., Iglesias-Suarez, F., Shamekh, S., Gentine, P., Giorgetta, M. A., Pincus, R., and Eyring, V.: Interpretable machine learning-based radiation emulation for ICON, Journal of Geophysical Research: Machine Learning and Computation, 2, e2024JH000501, 2025.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

He, K., Zhang, X., Ren, S., and Sun, J.: Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NY, USA, pp. 770-778, 2016.

Hernandez-Orallo, J.: Explainable AI: interpreting, explaining and visualizing deep learning, Computing reviews, (1):62, 2021.

Hogan, R. and Bozzo, A.: ECRAD: A new radiation scheme for the IFS, ECMWF Technical Memorandum number 787, 2016.

Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., and Collins, W. D.: Radiative forcing by long-lived greenhouse gases: calculations with the AER radiative transfer models, Journal of Geophysical Research Atmospheres, vol. 113, no. 13, D13103, 2008.

Jing, H.: Machine learning emulation for radiation parameterization (online version), https://doi.org/10.5281/zenodo.19249797, 2026.

Krasnopolsky, V., Fox-Rabinovitz, M., and Chalikov, D.: New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model, Monthly Weather Review, 133(5): 1370-1383, 2005.

Liang, J., Koji, T., and Takemasa, M.: A machine learning approach to the observation operator for satellite radiance data assimilation, Journal of the Meteorological Society of Japan Ser II, 101(1), 2023.

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., Smith, T., and Williams, J. K.: Using Artificial Intelligence to Improve Real-Time Decision-Making for High-Impact Weather, Bulletin of the American Meteorological Society, 98, 2073–2090, 2017.

Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., and Clough, S. A.: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave, Journal of Geophysical Research, 102, 16663-16682, 1997.

Morcrette, J. J., Barker, H. W., Cole, J. N. S., Iacono, M. J., and Pincus, R.: Impact of a new radiation package, McRad, in the ECMWF integrated forecasting system, Monthly Weather Review, vol. 136, no. 12, 2008.

Mu, B., Chen, L., Yuan, S., and Qin, B.: A radiative transfer deep learning model coupled into wrf with a generic fortran torch adaptor, Frontiers in Earth Science, 11: 1149566, 2023.

Niu, Z., and Zou, X.: A machine learning approach for deriving atmospheric temperatures and typhoon warm cores from FY-3E MWTS-3 observations, Journal of Geophysical Research: Machine Learning and Computation, 1, e2024JH000170, 2024.

Oreopoulos, L., and Barker, H.: Accounting for subgrid-scale cloud variability in a multi-layer 1D solar radiative transfer algorithm, Quarterly Journal of the Royal Meteorological Society, 126, 301-330, 1999.

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., and Baldi, P.: A fortran-keras deep learning bridge for scientific computing, Scientific Programming, 1-13, 2020.

https://doi.org/10.5194/egusphere-2026-1735
Preprint. Discussion started: 7 August 2026

Price, E., Mielikainen, J., Huang, M., Huang, B., Huang, H. A., and Lee, T.: GPU-Accelerated Longwave Radiation Scheme of the Rapid Radiative Transfer Model for General Circulation Models (RRTMG), IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 7(8):3660-3667, 2014.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat: Deep learning and process understanding for data-driven earth system science, Nature, 566(7743): 195-204, 2019.

Schmidhuber, J.: Deep learning in neural networks: An overview, Neural networks, 61:85-117, 2015.

Song, H., and Roh, S.: Improved weather forecasting using neural network emulation for radiation parameterization, Journal of Advances in Modeling Earth Systems, 13.10, 2021.

Toms, B., Barnes, E. A., and Ebert-Uphoff, I.: Physically interpretable neural networks for the geosciences: Applications to earth system variability, Journal of Advances in Modeling Earth Systems, 12(9): e2019MS002002, 2020.

Ukamaka, O., Enihe, R., and Jubril, A.: Cloud Detection in Satellite Imagery Using Deep Convolutional Neural Networks, International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), Omu-Aran, Nigeria, pp. 1-6, 2024.

Ukkonen, P.: Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer, Journal of Advances in Modeling Earth Systems, 14.4, 2022.

Wang, X., Han, Y., Xue, W., Yang, G., and Zhang, G.: Stable climate simulations using a realistic general circulation model with neural network parameterizations for atmospheric moist physics and radiation processes, Geoscientific Model Development, 15(9):3923-3940, 2022.

Woo, S., Park, J., Lee, J., and Kweon, I. S.: CBAM: Convolutional block attention module, Proceedings of the European conference on computer vision (ECCV), 3-19, 2018.

Wu, Y., Geng, X., Liu, Z., and Shi, Z.: Tropical Cyclone Forecast Using Multitask Deep Learning Framework, IEEE Geoscience and Remote Sensing Letters, 19, 1–5, 2022.

Yuval, J., Gorman, P. O., and Hill, C.: Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision, Geophysical Research Letters, 48(6): e2020GL091363, 2021.

Zhong, X., Chen, L., Li, H., Buizza, R., Liu, J., Feng, J., Zhu, Z., Fan, X., Dai, K., Luo, J., Wu, J., and Lu, B.: FuXi-ENS: A machine learning model for efficient and accurate ensemble weather forecasting. Science Advances, 11(44), 2025.